



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/682,441	10/09/2003	Gunnlaugur Jonsson	3165.1000-000	9803

21005 7590 02/05/2008
HAMILTON, BROOK, SMITH & REYNOLDS, P.C.
530 VIRGINIA ROAD
P.O. BOX 9133
CONCORD, MA 01742-9133

EXAMINER

BASEHOAR, ADAM L

ART UNIT	PAPER NUMBER
----------	--------------

2178

MAIL DATE	DELIVERY MODE
-----------	---------------

02/05/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/682,441

Applicant(s)

JONSSON, GUNNLAUGUR

Examiner

Adam L. Basehoar

Art Unit

2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 01 November 2007.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-23,55-72 and 88-100 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-23,55-72 and 88-100 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
 - ☐ Certified copies of the priority documents have been received in Application No. _____.
 - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to the RCE filed 11/01/07.
2. All previous rejections to the claims are maintained.
3. Claims 1-23, 55-72, and 88-100 are remain pending in the case. Claims 1, 9, 16, 23, 55, 61, 67, 72, 88, 91, 94, and 97 are independent claims.

Claim Objections

4. The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: The phrase "computer usable medium," is not found to have proper antecedent basis in the specification, however it is necessary to use this terminology in order to properly define the claim within the boundaries of statutory subject matter, because the phrase "computer usable medium," appears to only reasonably convey hardware storage and forms of portable, physical article media to one of ordinary skill in the art (Paragraph 203: i.e. memories 1004 and 1008). In order to overcome the objection, an amendment to the specification is necessary constituting a non-exhaustive statement of what the phrase "computer usable medium" would be as it would have been known to one of ordinary skill in the art at the time of the invention, in order to verify that the term "computer usable medium," could not be taken in the context of non-statutory subject matter.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 16-23, 67-72, and 94-97 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. In regard to independent claims 16, 23, 67, 72, 94, and 97, the claims are considered non-statutory because the system and apparatus could be considered merely a software system or apparatus (Specification: Paragraph 203: "software implementation of the invention"). Where there is no evidence in the specification that a term or phrase which may be interpreted as software, hardware or combinations thereof necessarily includes hardware, it should be interpreted in its broadest reasonable sense as software. The stated system and apparatus must be claimed in combination with appropriate hardware to enable the functionality of the system and apparatus to be realized. Appropriate correction required.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-4, 7, 9-11, 14, 16-18, 21, 23, and 98-100 are rejected under 35 U.S.C. 103(a) as being unpatentable over Alden et al (US-2002/0078086 06/20/02) in view of Cahill et al (US-7,099,890 08/29/06).

-In regard to substantially similar independent claims 1, 9, 16, and 23, Alden teaches
A method of developing computer software from an electronic spreadsheet, the method comprising the computer implemented steps of: coupling content/instructions in at least one cell of an electronic spreadsheet having an input cell to a window, where the content in the cell includes instruction for the object (Paragraph 47: “the data entity....calculation entity”; Paragraphs 49 and 50)(Fig. 10: 122 and 124); processing the content associated with the cell in the electronic spreadsheet; and determining properties of the window based on the content in the cell of the spreadsheet (Paragraphs 9-10: “identifying one or more cells in the spreadsheet....each entity corresponds to one or more cells”)(Figs. 8 & 9).

Alden does not specifically teach creating an object-oriented programming environment via said spreadsheet and representing an object in an object-oriented programming language as an electronic spreadsheet, where the electronic spreadsheet is the object and using an object-oriented method defined in at least one cell of the electronic spreadsheet, enabling the object to communicate with another object. Cahill et al teach creating an object-oriented programming environment via said spreadsheet and representing an object in an object-oriented programming language as an electronic spreadsheet (column 1, lines 57-67; column 2, lines 10-22, column 5, lines 36-61: “object can be embedded in a cell of a spreadsheet”), where the electronic spreadsheet is the object and using an object-oriented method defined in at least one cell of the electronic spreadsheet, enabling the object to communicate with another object (column 6, lines 15-20: “the spreadsheet object 306 can be a stand-alone spreadsheet document”; column 5, lines 35-61: “provides a means for brokering between objects that are designed to communicate with one another...object can communicate with another object through the interface....an object can

be embedded in a cell of a spreadsheet component”; column 2, lines 1-26). It would have been obvious to one of ordinary skill in the art at the time of the invention for Alden to have incorporated the object-oriented functionality as shown in Cahill, because Cahill taught that spreadsheets were ultimately a programming tool and that object oriented programming had ever-expanding popularity (column 1, lines 25-35) and that by providing said functionality a user could gain the advantage of being able to access the functionality of external objects (column 2, lines 6-10 & 17-22: “Advantageously, the spreadsheet component...provides sophisticated program environment...object can be enhanced...invoke the functionality of external objects”).

-In regard to dependent claims 2, 10, and 17, Alden teaches wherein the step of coupling content in at least one cell of an electronic spreadsheet to a window further includes determining any graphical or functional attributes associated with the content in the cell to construct the window (Paragraphs 9-10: “automatically updating” and “predetermined attributes....to determine configuration of the entities”).

-In regard to dependent claims 3, Alden teaches wherein the content in the cell of the spreadsheet corresponds to the properties of the window in that any changes to the properties of the window are reflected in the content in the cell (Paragraphs 9-10: “detects any change in the cells of the spreadsheet or any change in the entities of the influence diagram....influence diagram.”)(Fig. 10).

-In regard to dependent claims 4, 11, and 18, Alden teaches wherein the step of coupling the content is in response to determining that an event has occurred (Paragraphs 9-10 & 53-54: “a response to already existing spreadsheet” and “detecting any changes in cells....influence diagram”).

-In regard to dependent claims 7, 14, and 21, Alden teaches wherein the content in one of the cell includes any attributes associated with the cell (Paragraph 10: “identifying one or more cells....data or calculation cells....predetermined attributes”)(Note: While not relied upon, Cahill et al also teach said feature wherein the content of the cell includes any attributes associated with the cell as shown in Figures 3 and 4).

-In regard to dependent claims 98, 99, and 100, Alden teaches enabling an event to process changes and maintain equivalence events for the cells of spreadsheet (Fig. 10: 132 and 134). Alden does not specifically teach where the event includes an event type, an event handler, and event target. Cahill et al teach enabling an event to be defined in one of the cells of one of the spreadsheets (column 1, lines 57-67; column 5, lines 55-61), where the event includes an event type (column 1, lines 57-67; column 5, lines 55-61: i.e. object data type), an event handler (column 6, lines 25-61: i.e. recalculation component in view of other external objects through the component management system; column 7, lines 34-60)(Fig. 3: 304), and event target (column 7, lines 11-17: i.e. external object). It would have been obvious to one of ordinary skill in the art at the time of the invention for Alden to have incorporated the event, event handler, and event target of Cahill, because Cahill taught that object oriented programming had ever-expanding

popularity (column 1, lines 25-35) and that by providing said functionality a user could gain the advantage of being able to access the functionality of external objects (column 2, lines 6-10 & 17-22) via event objects stored embedded in the cells of a spreadsheet (column 5, lines 55-61).

9. Claim 55-59, 61-65, 67-70, 72, and 88-97 rejected under 35 U.S.C. 103(a) as being unpatentable over Microsoft, Excel 2000 Screenshots, 12/31/99, pp. 1-15 (Hereafter Excel 2000) in view of Cahill et al (US-7,099,890 08/29/06).

-In regard to substantially similar independent claims 55, 61, 67, and 72, Excel 2000 teaches a method of providing text editor functionality in an electronic spreadsheet, comprising: defining a code column to program the computer software in the electronic spreadsheet (Page 7: i.e. user defines column "A"); and responding to a request for a new line by inserting a new cell in the code column (Pages 8 & 9: i.e. user requests a new line by inserting a new cell into the column "A").

Excel 2000 does not specifically teach wherein the created spreadsheet provided a software programming environment using the electronic spreadsheet in which software code was programmed, and where the spreadsheet corresponds to an object in an object-oriented programming language, the object communicating with another object using a method that was specified in a cell of the spreadsheet. Cahill et al teach creating an object-oriented programming environment via an electronic spreadsheet in which software code was programmed in the spreadsheet (column 1, lines 25-39 & 57-67: "spreadsheet is ultimately a programming tool"; column 2, lines 10-34: "spreadsheet components make available a new class of object functions"; column 5, lines 36-61: "object can be embedded in a cell of a spreadsheet")(Fig. 4), where the

electronic spreadsheet is the object and using an object-oriented method defined in at least one cell of the electronic spreadsheet, enabling the object to communicate with another object (column 6, lines 15-20: "the spreadsheet object 306 can be a stand-alone spreadsheet document"; column 5, lines 35-61: "provides a means for brokering between objects that are designed to communicate with one another...object can communicate with another object through the interface....an object can be embedded in a cell of a spreadsheet component"; column 2, lines 1-26). It would have been obvious to one of ordinary skill in the art at the time of the invention for Excel 2000 to have incorporated the object-oriented functionality as shown in Cahill, because Cahill taught that spreadsheets were ultimately a programming tool and that object oriented programming had ever-expanding popularity (column 1, lines 25-35) and that by providing said object function functionality a user could gain the advantage of being able to access the functionality of external objects (column 2, lines 6-10 & 17-22: "Advantageously, the spreadsheet component...provides sophisticated program environment...object can be enhanced...invoke the functionality of external objects").

-In regard to dependent claims 56, 62, and 68, Excel 2000 teaches wherein the code column further includes text editor functionality in that the code column behaves as a text editor (Page 10: i.e. sample text in code column "A" is provided a plurality of text editing functions such as spell check, font selection, justification, etc).

-In regard to dependent claims 57, 63, and 69, Excel 2000 teaches wherein the request for a new line occurs when a keystroke input is received while a cell in the code column is

selected (Pages 8 & 9: i.e. A4 was selected in code column "A" for inserting a new row or shifting the cells down to insert a new cell via a keystroke).

-In regard to dependent claims 58, 64, and 70, Excel 2000 further includes: inserting the new cell at the selected cell (Pages 12 & 13: i.e. shows inserting a new cell at line A6 of the column "A"); moving cells positioned below the selected cell to create space for the new cell (Pages 12 & 13: i.e. the code in cell "A7" was moved down to "A8"); and correcting any references to the cells which are moved (Pages 11 & 13: i.e. before the added cell "A7" referenced "A5" and "A6", but after the added cell, the references were corrected to "A5" and "A7"). Excel 2000 does not specifically teach wherein the new cell was added below the selected cell. It would have been obvious to one of ordinary skill in the art at the time of the invention for Excel 2000 to have allowed the insertion of cells below the selected cell, because it was notoriously well known in the art at the time of the invention that providing the new cell below the selected cell, instead of at the selected cell, would provide the similar benefit of allowing the user to input additional data into a new cell of the selected column.

-In regard to dependent claims 59 and 65, Excel 2000 teaches wherein the code column further includes scroll bars (Pages 14 & 15: i.e. inserting a scroll bar into the code column "A").

-In regard to substantially similar independent claims 88, 91, 94, and 97, Excel 2000

teaches a method, apparatus, and system of programming with an electronic spreadsheet, the method comprising:

defining cells in a spreadsheet that are associated with an iterative process (Page 6: i.e. cells A1→A3) repeating for one or more cycles (Page 3: i.e. for four cycles), where the iterative process repeats the same action until a condition no longer applies (Pages 2→6: i.e. when the maximum iterations equals four times the iterative processing of the calculation no longer applies and the iterative processing ceases); and at each cycle, determining whether to modify content in the cells associated with the iterative process (Pages 4→5: i.e. A1 initially set to 5 +A3, A2 initially set to 10, A3 initially set to A1+A2+A3 with the final result being the values 110, 10, and 225 on Page 6).

Excel 2000 does not specifically teach wherein the created spreadsheet provided a software programming environment using the electronic spreadsheet in which a software program was programmed and where resulting modified content causes changes to the software program being created using the spreadsheet, and where the spreadsheet corresponds to an object in an object-oriented programming language, the object communicating with another object using a method that was specified in a cell of the spreadsheet. Cahill et al teach creating an object-oriented programming environment via an electronic spreadsheet in which a software program was programmed in the spreadsheet (column 1, lines 25-39 & 57-67: "spreadsheet is ultimately a programming tool"; column 2, lines 10-34: "spreadsheet components make available a new class of object functions"; column 5, lines 36-61: "object can be embedded in a cell of a spreadsheet")(Fig. 4) and where resulting modified content causes changes to the software

program being created using the spreadsheet (column 6, lines 25-67; column 7, lines 1-67: i.e. recalculation component is responsible for causing the spreadsheet object to recalculate the value of the cells...whenever the spreadsheet object has changed)(Figs. 3 and 4), where the electronic spreadsheet is the object and using an object-oriented method defined in at least one cell of the electronic spreadsheet, enabling the object to communicate with another object (column 6, lines 15-20: "the spreadsheet object 306 can be a stand-alone spreadsheet document"; column 5, lines 35-61: "provides a means for brokering between objects that are designed to communicate with one another...object can communicate with another object through the interface....an object can be embedded in a cell of a spreadsheet component"; column 2, lines 1-26). It would have been obvious to one of ordinary skill in the art at the time of the invention for Excel 2000 to have incorporated the object-oriented functionality as shown in Cahill to generate modified program code, because Cahill taught that spreadsheets were ultimately a programming tool and that object oriented programming had ever-expanding popularity (column 1, lines 25-35) and that by providing said object function functionality a user could gain the advantage of being able to access the functionality of external objects (column 2, lines 6-10 & 17-22: "Advantageously, the spreadsheet component...provides sophisticated program environment...object can be enhanced...invoke the functionality of external objects") by utilizing object function in spreadsheet cells to create modifiable software programs (Figs. 3 and 4).

-In regard to dependent claims 89, 92, and 95, Excel 2000 teaches wherein at least one of the cells associated with the iterative process includes a final value cell (Page 6: $A3 = A1 + A2 + A3$), and at least one of the cells includes an initial value cell (Page 4: $A1 = 5 + A3$),

where a value in the final value cell is used to modify a value in the initial value cell (Page 4: i.e. the result determined in A3 was utilized to calculate the value of A1).

-In regard to dependent claim 90, 93, and 96, Excel 2000 teaches wherein the iterative process is repeated for either a fixed number of times or until a condition defined in a condition cell no longer applies, or begins to apply (Pages 2→3: i.e. Maximum Iterations equals four times).

10. Claims 5-6, 8, 12-13, 15, 19-20, and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Alden et al (US-2002/0078086 06/20/02) in view of Cahill et al (US-7,099,890 08/29/06) in view of Duane et al (US-6,243,721 06/05/01).

-In regard to dependent claims 5, 12, and 19, Alden teaches wherein the step of determining that an event has occurred includes determining (Paragraph 53: “changes in the spreadsheet or in the equivalent visual representation.”). Alden does not specifically teach wherein the event included a drag and drop event dragging the content cell from the spreadsheet and dropping it onto the window. Duane et al teach detecting a drag and drop event wherein the content item was dragged and dropped onto a window (column 3, lines 9-23; column 7, lines 19-27)(Fig. 5: “Step 1”). It would have been obvious to one of ordinary skill in the art at the time of the invention for the determining of changes of Alden to have included determining a drag and drop as taught in Duane et al, because Duane et al teaches that dragging and dropping a content item onto a window provides the benefit of increased efficiency and reduced complexity by automatically positioning, aligning, and binding content objects in a window which would have

increased efficiency of creating the influence diagram of Alden (column 3, lines 9-23; column 7, lines 19-27).

-In regard to dependent claims 6, 13, and 20, Alden further includes the step of responding to the dropping of the content from the cell onto the window by: (a) processing the content in the cell (Fig. 10: 122, 124); and (b) determining the properties of the window based on the content in the cell including determining any desired behavior of the object including any desired appearance of the window to depict the object based on the content in the cell (Paragraphs 9-10 & 51-52)(Fig. 10: 126, 128, 130).

-In regard to dependent claims 8, 15, and 22, Alden teaches wherein the predetermined attributes include shape, color, border, style, image, etc (Paragraph 53). Alden does not specifically teach wherein the predetermined attributes included any one of a plurality of input controls. Duane et al teach wherein the attributes could include the input controls of labels, text boxes, combo boxes, list boxes, etc (column 11, lines 40-46). It would have been obvious to one of ordinary skill in the art at the time of the invention for the attributes to have included any input control as shown in Duane et al, because Duane et al teach that the automatic placement of control items provides the benefit of increased efficiency and reduced complexity by automatically positioning, aligning, and binding content objects in a window which would have increased efficiency of creating the influence diagram of Alden (column 3, lines 9-23; column 7, lines 19-27). In addition, the Examiner notes that it was notoriously well known in the art at the

time of the invention for the cells of spreadsheets to have included user defined input controls to provide the benefit of additional data manipulation functionality to spreadsheets.

11. Claims 60, 66, and 71, are rejected under 35 U.S.C. 103(a) as being unpatentable over Microsoft, Excel 2000 Screenshots, 12/31/99, pp. 15 (Hereafter Excel 2000) in view of Cahill et al (US-7,099,890 08/29/06) in further view of Stead (US-6,690,401 02/10/04).

-In regard to dependent claims 60, 66, and 71, Excel 2000 does not specifically teach wherein the scroll bars enable scrolling through the code column independent of any scrolling of the spreadsheet. Stead teaches vertical cell scrolling for scrolling individual columns of a spreadsheet in addition to the conventional spreadsheet scrolling (column 5, lines 46-63)(Figs. 1 & 9). It would have been obvious to one of ordinary skill in the art at the time of the invention for the scroll bars associated with the code column in Excel 2000 to have allowed independent column scrolling, because Stead teaches that vertical cell scrolling provides the benefit of allowing a greater number of columns of data to be displayed within a window/display area such as those of data columns in a spreadsheet (column 4, lines 14-28).

Response to Arguments

12. Applicant's arguments filed 11/01/07 have been fully considered but they are not persuasive.

-In regard to the independent claims, Applicant argues that Cahill does not teach or suggest the newly added limitation wherein the electronic spreadsheet was the object in the object oriented programming language. The Examiner respectfully disagrees with the Applicant.

Cahill clearly teaches wherein the user defined spreadsheet object data type could be a stand-alone spreadsheet document (column 6, lines 15-16: "spreadsheet object 306 can be a stand-alone spreadsheet document").


Conclusion

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Adam L. Basehoar whose telephone number is (571)-272-4121. The examiner can normally be reached on M-F: 7:00am - 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Steve Hong can be reached on (571) 272-4124. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Adam L. Basehoar


02/01/08